

## **IN THE CLAIMS:**

1. (Currently amended) A method, in a data processing system, for automatically identifying performance regression between builds of a computer program based on trace data obtained from a plurality of executions of a first and second build of ~~[[a]]the~~ computer program, comprising:

obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the first and second builds of the computer program;

generating a minimized call tree data structure from the plurality of call tree data structures for each of the first and second builds of the computer program, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures, wherein generating the minimized call tree data structure further comprises:

copying a first call tree data structure for a selected build of the computer program; and  
walking a second call tree data structure for the selected build of the computer program  
over the first call tree data structure to generate the minimized call tree data structure, wherein  
walking the second call tree data structure over the first call tree data structure further comprises:

for each node that exists in both the first call tree data structure and the second  
call tree data structure, generating a node in the minimized call tree data structure and  
associating values with the node, wherein the values associated with the node are values  
that correspond to a minimum of the values associated with corresponding nodes in the  
first call tree data structure and the second call tree data structures; and

for each node that exists in only one of the first call tree data structure and the  
second call tree data structure, inhibiting creating a node in the minimized call tree data  
structure;

subtracting the minimized call tree data structure for the second build of the computer program from the minimized call tree data structure of the second computer program to thereby generate a subtracted minimized call tree data structure, wherein subtracting the minimized call tree data structure  
for the second build from the minimized call tree data structure for the first build to generate a subtracted  
minimized call tree data structure further comprises:

copying the minimized call tree data structure for the first build; and  
walking the minimized call tree data structure for the second build over the minimized  
call tree data structure for the first build to generate the subtracted minimized call tree data  
structure, wherein walking the minimized call tree data structure for the second build over the  
minimized call tree data structure for the first build further comprises:

for each node that exists in both the minimized call tree data structure for the first build and the minimized call tree data structure for the second build, generating a node in the subtracted minimized call tree data structure by subtracting a minimum base value of the node in the minimized call tree data structure for the second build from a minimum base value of a corresponding node in the minimized call tree data structure for the first build; and

for each node that exists in only one of the minimized call tree data structure for the first build and the minimized call tree data structure for the second build, creating a node in the subtracted minimized call tree data structure having a negative minimum base value corresponding to a minimum base value of the node that exists in either of the minimized call tree data structure for the first build or the minimized call tree data structure for the second build; [[and]]

outputting the subtracted minimized call tree data structure; and

inputting the trace data to an arcflow tool, wherein the arcflow tool generates the plurality of call tree data structures for each of the first and second builds of the computer program based on the trace data.

2.-20. (Canceled)